UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

SECCIÓN: CC52

GRUPO: 3

CURSO: Fundamentos de Data Science

PROFESOR(A): Nérida Isabel Manrique Tunque

TÍTULO: Trabajo Parcial Fundamentos de Data Science

El presente trabajo ha sido realizado por:

Joaquín Eduardo Velarde Leyva	U202212510
Victor Daniel Chipana Gutierrez	U202115805
Daniel Ivan Carbajal Robles	U20221B751
Ian Joaquin Sanchez Alva	U202124676

2024-01

Índice:

I. Introducción

El análisis exploratorio de datos (EDA) es una etapa crucial en cualquier proyecto de análisis de datos. En este informe, se llevará a cabo un EDA utilizando RStudio como herramienta de software sobre el conjunto de datos "Hotel booking demand".

El objetivo de este análisis exploratorio de datos es comprender en profundidad el conjunto de datos "Hotel booking demand" a través de técnicas de visualización, preparación y análisis en RStudio. Se busca identificar patrones y tendencias en las reservas de hotel, evaluar la calidad de los datos abordando valores faltantes y datos atípicos, explorar relaciones entre variables como la duración de la estadía y la disponibilidad de estacionamiento, comparar el comportamiento entre el hotel urbano y el resort, y obtener insights básicos para la toma de decisiones estratégicas en la gestión hotelera.

II. Caso de análisis

El origen de los datos se encuentra en diversas fuentes, como las reservas de habitaciones, las reseñas de huéspedes, los datos de ventas y las estadísticas de ocupación. Los datos se recolectan a través de diferentes herramientas y sistemas, como las reservas centrales, los sistemas de gestión de reservas y los sistemas de gestión de datos. La recopilación de datos fue realizada por Nuno Antonio, Ana de Almeida y Luis Nunes

Diccionario de variables:

Nombre	Tipo	Descripción
hotel	character	Nombre de los hoteles (Resort Hotel, City Hotel)
is_canceled	integer	Indica si se cancela la reserva (0 no se canceló, 1 se canceló)

lead_time	integer	Número de días transcurridos entre la fecha de reserva con la fecha de llegada
arrival_date_year	integer	Año de fecha de llegada
arrival_date_month	character	Mes de fecha llegada
arrival_date_week_number	integer	Número de semana de la fecha llegada
arrival_date_day_of_month	integer	Día del mes de la fecha de llegada
stays_in_weekend_nights	integer	Número de noches que el huespede se quedó o reservó en el hotel para fin de semana
stays_in_week_nights	integer	Número de noches que el huespede se quedó o reservó en el hotel entre semana
adults	integer	Número de adultos
children	integer	Número de niños
babies	integer	Número de bebes
meal	character	Tipo de comida reservada (BB, FB, HB, SC, Undefined)
country	character	País de origen
market_segment	character	Designación del segmento de mercado
distribution_channel	character	Canal de distribución de cocina
is_repeated_guest	integer	Valor que indica si el nombre de la reserva fue de alguien repetido (1 -> si, 0 -> no)
previous_cancellations	integer	Número de reservas anteriores que fueron cancelados por el cliente antes de la reserva actual
previous_bookingsd_not_canceled	integer	Número de reservas anteriores no cancelados por

		el cliente antes de la reserva actual
reserved_room_type	character	Código del tipo de habitación de reserva
assigned_room_type	character	Código del tipo de habitación asignado al reservar
booking_changes	integer	Número de cambios realizadas a la reserva desde el momento en que la reserva se realizó hasta el momento de cancelación o entrega
deposit_type	character	Indicación si el cliente realizó un depósito para garantizar la reserva
agent	character	ID de la agencia de viajes que realizó la reserva
company	character	ID de la compañía que realizó la reserva o responsable del pago de la reserva
days_in_waiting_list	integer	Número de días que la reserva estuvo en la lista de espera antes de confirmarse por el cliente
customer_type	character	Tipo de reserva(contract, group, transient, transient-party)
adr	numeric	Tarifa diaria promedio
required_car_parking_spaces	integer	Número de plazas de aparcamiento necesarias para el cliente
total_of_special_request	integer	Número de solicitudes especiales realizadas por el cliente
reservation_status	character	Último estado de la reserva (Canceled, Check-Out, No-Show)

reservantion_status_date	date	Fecha en la que se cambió el último estado

La herramienta utilizada fue RStudio, software que usa el lenguaje para análisis de datos “R”. Gracias a las funciones de las librerías, que se pueden descargar en el software, tiende a facilitar el análisis de los datasets. Las librerías te ayudan a facilitar diversas tareas. Por ejemplo, ggplot2 es una librería utilizada para graficar, y otras más.

Los casos de uso aplicable son:

- Predicción de Cancelaciones: El caso principal es predecir la probabilidad de que una reserva de hotel sea cancelada.
- Pronóstico de Demanda: Utilizando los datos históricos de reservas, se puede pronosticar la demanda de habitaciones de hotel para fechas futuras, asimismo, el tipo de reserva que se hace como, cantidad de adultos, niños, bebes, si pagó un adelanto, etc.
- Análisis de Mercado: Al comparar el rendimiento del hotel con el de la competencia y analizar las tendencias del mercado.

Desarrollo de preguntas de análisis:
Primero para responder estas preguntas se hizo lo siguiente:

Lectura de los datos desde la ruta (Los espacios vacíos fueron reemplazos por NA):

```
setwd("C:/Users/Usuario/Desktop/Querys de R")
hotels<- read.csv('hotel_bookings.csv', header = TRUE , sep = ',', dec
= '.',stringsAsFactors = FALSE , na.strings = "")
```

Limpieza de datos (Se utiliza ese comando para poder omitir los NA y tener una data limpia):

```
hotels_data.limpia <- na.omit(hotels)
```

Y se usaron estas librerías:

```
install.packages("ggplot2")
install.packages("dplyr")
library(ggplot2)
library(dplyr)
```

A. ¿Cuántas reservas se realizan por tipo de hotel? o ¿Qué tipo de hotel prefiere la gente?

```
##Este data frame solo tiene los hoteles y las reservas
datos <- data.frame(
  hotels_data.limpia$hotel ,
  hotels_data.limpia$is_canceled
)
```

head(datos)

```
 hotels_data.limpia.hotel hotels_data.limpia.is_canceled
1             Resort Hotel                              0
2             Resort Hotel                              0
3             Resort Hotel                              0
4             Resort Hotel                              0
5             Resort Hotel                              0
6             Resort Hotel                              0
```

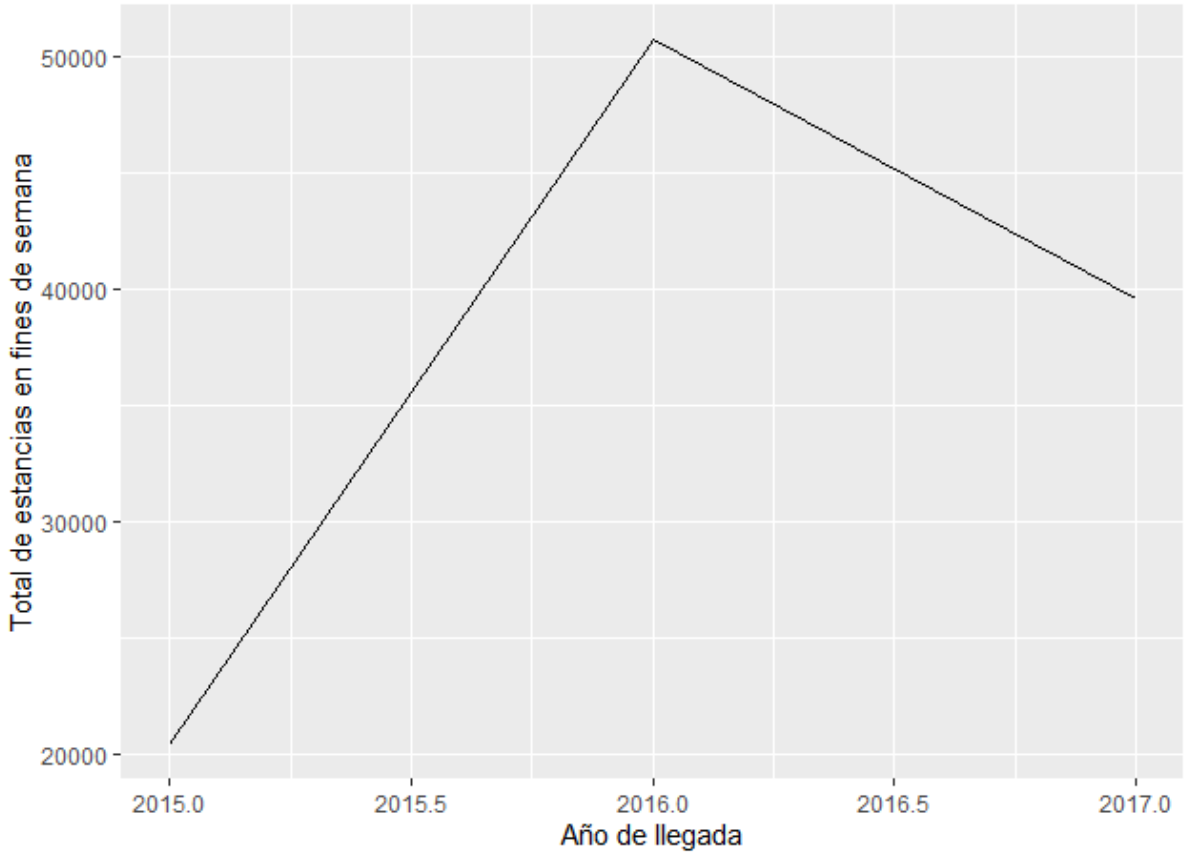
```
conteo_por_hotel <- table(datos$hotels_data.limpia.hotel)
conteo_por_hotel
City Hotel Resort Hotel
       79330        40060
```

Con esta tabla “conteo_por_hotel” podemos responder a la pregunta de cuántas reservas se realizan por hotel. Y concluimos que en el tipo de hotel “City Hotel” se realizaron 79330 reservas y en “Resort Hotel” 40060.

```
barplot(conteo_por_hotel,
        main = "Preferencia de hotel",
        xlab = "Tipo de hotel",
        ylab = "Número de reservas",
        col = "#CCBC0F",
        border = "black")
```

Con este gráfico se puede observar la diferencia y también concluir que la gente prefiere el tipo “City Hotel”

```
demanda_por_anio <- summarise(datos_agrupados,
total_stays_in_weekend_nights = sum(stays_in_weekend_nights))

arrival_date_year total_stays_in_weekend_nights
              <int>                         <int>
1              2015                         20450
2              2016                         50695
3              2017                         39601

ggplot(demanda_por_anio, aes(x = arrival_date_year, y =
total_stays_in_weekend_nights)) +
  geom_line() +
  labs(x = "Año de llegada", y = "Total de estancias en fines de
semana")
```



Con el gráfico y la tabla de demanda_por_anio podemos concluir que la tendencia baja después del año 2016.

C. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?

```
## convertir los meses a factor para que puedan ser ordenados
hotels_data.limpia$arrival_date_month <-
factor(hotels_data.limpia$arrival_date_month, levels =
c("January", "February", "March", "April", "May", "June",
"July", "August", "September", "October", "November",
"December"))
```

```
##Se hace un conteo  de las reservas por mes
reservas_por_mes <- table
(hotels_data.limpia$arrival_date_month)
```

```
 January  February     March     April       May      June      July    August September   October  November  December
    5929      8068      9794     11089     11791     10939     12661     13877     10508     11160      6794      6780
```

```
## sacamos el promedio de la tabla anterior para cada mes
promedio_reservas <- mean(reservas_por_mes)
promedio_reservas
[1] 9949.167

## le damos valores para definir las temporadas
temporada_alta <- promedio_reservas * 1.2  ## se usó 1.2 y 0.8
para poder definir lo que es una temporada baja y alta
temporada_baja <- promedio_reservas * 0.8  ## si es más del 20%
abajo del promedio o si es más del 20% arriba del promedio.
> temporada_baja
[1] 7959.333
> temporada_alta
[1] 11939

## se le da los valores de las temporadas a los meses
temporada <- cut(reservas_por_mes, breaks = c(-Inf,
temporada_baja, temporada_alta, Inf), labels = c("baja",
"media", "alta"))
##inf y -inf son necesarios en la funcion cut porque indican de
donde a donde van los valores de los promedios de las reservas

temporada (de todos los meses)
 [1] baja  media media media media media alta  alta  media media
baja  baja
Levels: baja media alta

# resumen final de los meses con su número de reservas totales y
la temporada a la que pertenecen
resumen_temporadas <- data.frame(Mes = names(reservas_por_mes),
Total_Reservas = as.numeric(reservas_por_mes), Temporada =
temporada)
print (resumen_temporadas)
```

	Mes	Total Reservas	Temporada
1	January	5929	baja
2	February	8068	media
3	March	9794	media
4	April	11089	media
5	May	11791	media
6	June	10939	media
7	July	12661	alta
8	August	13877	alta
9	September	10508	media

```
10   October          11160          media
11  November           6794          baja
12  December           6780          baja
```

Así se puede conocer (con los valores que hemos dado) cuales son las temporadas bajas, medias y altas

D. ¿Cuándo es menor la demanda de reservas?

```
## utilizando la pregunta anterior , filtramos solo con los
meses de temporada baja
meses_temporada_baja <-
resumen_temporadas[resumen_temporadas$Temporada == "baja", ]
print(meses_temporada_baja)
```

```
    Mes Total_Reservas Temporada
1   January           5929      baja
11 November           6794      baja
12 December           6780      baja
```

Con esta tabla se puede concluir que los meses donde de reservas es baja es en enero, noviembre, diciembre.

E. ¿Cuántas reservas incluyen niños y/o bebes?

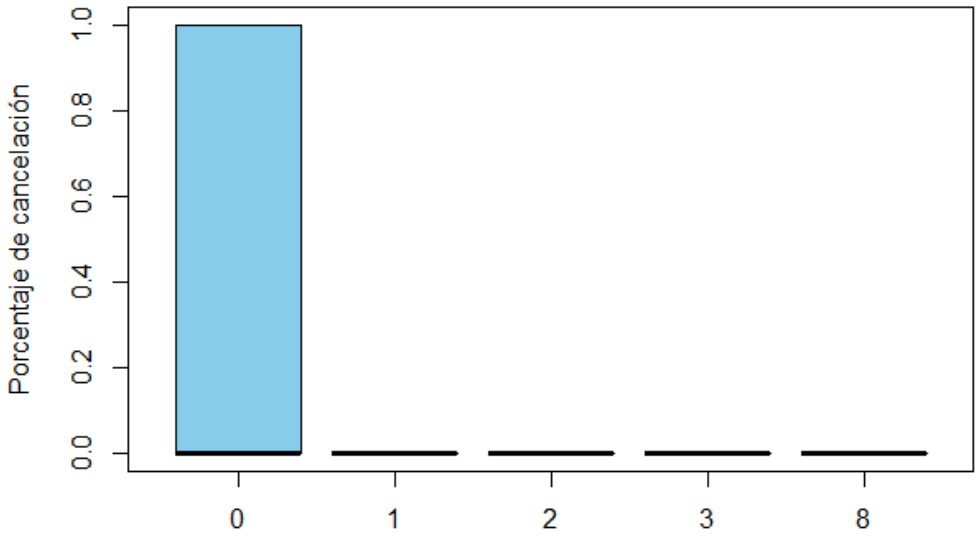
```
## filtramos las filas donde hayan niños o bebés
reservas_con_ninos_o_bebes <-
hotels_data.limpia[hotels_data.limpia$children > 0
|hotels_data.limpia$babies > 0, ]
##conteo
total_reservas_con_ninos_o_bebes <-
nrow(reservas_con_ninos_o_bebes)
print(total_reservas_con_ninos_o_bebes)
[1] 9336
```

Con este filtrado de los datos y conteo de estos mismo podemos determinar cuántas reservas incluyen niños.

F. ¿Es importante contar con espacios de estacionamiento?

```
boxplot(hotels_data.limpia$is_canceled ~
hotels_data.limpia$required_car_parking_spaces, xlab="Espacios
de parque necesarios", ylab="Porcentaje de cancelación",
col="skyblue")
```

Este gráfico nos indica que los espacios de estacionamiento son importantes porque las cancelaciones de reservas se dan cuando los estacionamientos necesitados son 0

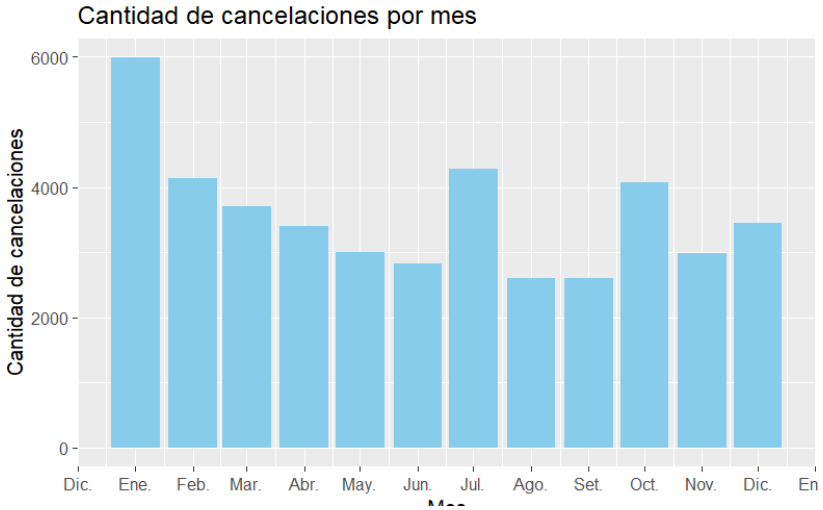


G. ¿

C

"(

```
cancelaciones$reservation_status_date <-
as.Date(cancelaciones$reservation_status_date)
cancelaciones$mes <-
format(cancelaciones$reservation_status_date, "%m")
cancelaciones_por_mes <- cancelaciones %>% group_by(mes) %>%
summarise(cantidad = n())
cancelaciones_por_mes$mes <- as.Date(paste("2000",
cancelaciones_por_mes$mes, "01", sep = "-"))
ggplot(cancelaciones_por_mes, aes(x = mes, y = cantidad)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  scale_x_date(date_labels = "%b", date_breaks = "1 month") +
  labs(title = "Cantidad de cancelaciones por mes", x = "Mes", y
= "Cantidad de cancelaciones")
```



Como se puede ver en la gráfica, la cantidad de cancelaciones se realizan por el mes de enero, dichos datos se obtuvieron a traves de las variables reservation_status y reservation_status_date

III. Conjunto de datos

Los datos que tenemos son los siguientes:

```
> str(hotels)
'data.frame':   119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ arrival_date_month            : chr  "July" "July" "July" "July" ...
 $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : chr  "0" "0" "0" "0" ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : chr  "BB" "BB" "BB" "BB" ...
 $ country                       : chr  "PRT" "PRT" "GBR" "GBR" ...
 $ market_segment                : chr  "Direct" "Direct" "Direct" "Corporate" ...
 $ distribution_channel          : chr  "Direct" "Direct" "Direct" "Corporate" ...
 $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : chr  "C" "C" "A" "A" ...
 $ assigned_room_type            : chr  "C" "C" "C" "A" ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : chr  "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
 $ agent                         : chr  "NULL" "NULL" "NULL" "304" ...
 $ company                       : chr  "NULL" "NULL" "NULL" "NULL" ...
 $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type                 : chr  "Transient" "Transient" "Transient" "Transient" ...
 $ adr                           : num  0 0 75 75 98 ...
 $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ total_of_special_requests     : int  0 0 0 0 1 1 0 1 1 0 ...
 $ reservation_status            : chr  "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
 $ reservation_status_date       : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
>
```

Como se puede visualizar, constamos de 32 tipos de datos en este dataset, así es como nos aseguramos de obtener la cantidad de variables para realizar la tabla de descripción de las variables.

Nombre	Descripción
hotel	Tipo de Hotel
is_canceled	valor que indica si la reserva fue cancelada o no
lead_time	Número de días transcurridos entre la fecha de reserva con la fecha de llegada
arrival_date_year	Año de fecha de llegada
arrival_date_month	Mes de fecha llegada
arrival_date_week_number	Número de semana de la fecha llegada
arrival_date_day_of_month	Día del mes de la fecha de llegada
stays_in_weekend_nights	Número de noches que el huespede se quedó o reservó en el hotel para fin de semana
stays_in_week_nights	Número de noches que el huespede se quedó o reservó en el hotel entre semana
adults	Número de adultos
children	Número de niños
babies	Número de bebes
meal	Tipo de comida agendada

country	País de origen
market_segment	Designación del segmento de mercado
distribution_channel	Canal de distribución de cocina
is_repeated_guest	Valor que indica si el nombre de la reserva fue de alguien repetido (1 -> si, 0 -> no)
previous_cancellations	Número de reservas anteriores que fueron cancelados por el cliente antes de la reserva actual
previous_bookingsd_not_canceled	Número de reservas anteriores no cancelados por el cliente antes de la reserva actual
reserved_room_type	Código del tipo de habitación de reserva
assigned_room_type	Código del tipo de habitación asignado al reservar
booking_changes	Número de cambios realizadas a la reserva desde el momento en que la reserva se realizó hasta el momento de cancelación o entrega
deposit_type	Indicación si el cliente realizó un depósito para garantizar la reserva
agent	ID de la agencia de viajes que realizó la reserva
company	ID de la compañía que realizó la reserva o responsable del pago de la reserva
days_in_waiting_list	Número de días que la reserva estuvo en la lista de espera antes de confirmarse por el cliente
customer_type	Tipo de reserva

adr	Tarifa diaria promedio
required_car_parking_spaces	Número de plazas de aparcamiento necesarias para el cliente
total_of_special_request	Número de solicitudes especiales realizadas por el cliente
reservation_status	Último estado de la reserva
reservantion_status_date	Fecha en la que se cambió el último estado

IV. Análisis Exploratorio de Datos

Carga de datos:

```
#Establece el directorio de trabajo (setwd) en la ruta especificada, que parece ser donde se encuentra el archivo "hotel_bookings.csv".
setwd("C:/Users/ASUS/OneDrive/Escritorio/Quinto ciclo/Fundamento_Data_Science/Trabajo_Parcial")

#Utiliza la función read.csv() para leer el archivo CSV llamado "hotel_bookings.csv". Los parámetros son:
df<-read.csv('hotel_bookings.csv', header=TRUE, sep=',',dec='.',stringsAsFactors = FALSE)
#header = TRUE: Indica que la primera fila del archivo CSV contiene nombres de columnas.
#sep = ',': Especifica que el delimitador de campo en el archivo CSV es una coma.
#dec = '.': Indica que el separador decimal en el archivo CSV es un punto.
#stringsAsFactors = FALSE: Evita que las cadenas de caracteres se conviertan automáticamente en factores.
```

Data

df 119390 obs. of 32 variables

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults
1	Resort Hotel	0	342	2015	July	27	1	0	0	2
2	Resort Hotel	0	737	2015	July	27	1	0	0	2
3	Resort Hotel	0	7	2015	July	27	1	0	1	1
4	Resort Hotel	0	13	2015	July	27	1	0	1	1
5	Resort Hotel	0	14	2015	July	27	1	0	2	2
6	Resort Hotel	0	14	2015	July	27	1	0	2	2
7	Resort Hotel	0	0	2015	July	27	1	0	2	2
8	Resort Hotel	0	9	2015	July	27	1	0	2	2
9	Resort Hotel	1	85	2015	July	27	1	0	3	2

children	babies	meal	country	market_segment	distribution_channel	is_repeated_guest	previous_cancellations	previous_bookings_not_canceled	reserved_room_type
0	0	BB	PRT	Direct	Direct	0	0	0	C
0	0	BB	PRT	Direct	Direct	0	0	0	C
0	0	BB	GBR	Direct	Direct	0	0	0	A
0	0	BB	GBR	Corporate	Corporate	0	0	0	A
0	0	BB	GBR	Online TA	TA/TO	0	0	0	A
0	0	BB	GBR	Online TA	TA/TO	0	0	0	A
0	0	BB	PRT	Direct	Direct	0	0	0	C
0	0	FB	PRT	Direct	Direct	0	0	0	C

assigned_room_type	booking_changes	deposit_type	agent	company	days_in_waiting_list	customer_type	adr	required_car_parking_spaces	total_of_special_requests
C	3	No Deposit	NULL	NULL	0	Transient	0.00	0	0
C	4	No Deposit	NULL	NULL	0	Transient	0.00	0	0
C	0	No Deposit	NULL	NULL	0	Transient	75.00	0	0
A	0	No Deposit	304	NULL	0	Transient	75.00	0	0
A	0	No Deposit	240	NULL	0	Transient	98.00	0	1
A	0	No Deposit	240	NULL	0	Transient	98.00	0	1
C	0	No Deposit	NULL	NULL	0	Transient	107.00	0	0
C	0	No Deposit	303	NULL	0	Transient	103.00	0	1
A	0	No Deposit	240	NULL	0	Transient	82.00	0	1

reservation_status	reservation_status_date
Check-Out	2015-07-01
Check-Out	2015-07-01
Check-Out	2015-07-02
Check-Out	2015-07-02
Check-Out	2015-07-03
Check-Out	2015-07-03
Check-Out	2015-07-03
Check-Out	2015-07-03
Canceled	2015-05-06

Inspeccionar datos:

- Verificar tipos de datos de las columnas y nombres de las columnas:

```
column_info <- function(hotels) {
  cat("Nombres de las columnas:\n")
  print(names(hotels))
  cat("\nTipos de datos de las columnas:\n")
  print(sapply(hotels, class))
}

column_info(hotels)
```

```
> column_info(df)
Nombres de las columnas:
 [1] "hotel"                     "is_canceled"               "lead_time"                 "arrival_date_year"
 [5] "arrival_date_month"        "arrival_date_week_number"  "arrival_date_day_of_month" "stays_in_weekend_nights"
 [9] "stays_in_week_nights"      "adults"                    "children"                  "babies"
[13] "meal"                      "country"                   "market_segment"            "distribution_channel"
[17] "is_repeated_guest"         "previous_cancellations"    "previous_bookings_not_canceled" "reserved_room_type"
[21] "assigned_room_type"        "booking_changes"           "deposit_type"              "agent"
[25] "company"                   "days_in_waiting_list"      "customer_type"             "adr"
[29] "required_car_parking_spaces" "total_of_special_requests" "reservation_status"        "reservation_status_date"
```

```
Tipos de datos de las columnas:
                         hotel                    is_canceled                      lead_time                  arrival_date_year
                   "character"                      "integer"                      "integer"                      "integer"
            arrival_date_month         arrival_date_week_number        arrival_date_day_of_month          stays_in_weekend_nights
                   "character"                      "integer"                      "integer"                      "integer"
          stays_in_week_nights                         adults                       children                         babies
                   "integer"                      "integer"                      "integer"                      "integer"
                        meal                        country                  market_segment             distribution_channel
                   "character"                    "character"                    "character"                    "character"
           is_repeated_guest          previous_cancellations previous_bookings_not_canceled                reserved_room_type
                   "integer"                      "integer"                      "integer"                    "character"
           assigned_room_type                booking_changes                   deposit_type                          agent
                   "character"                      "integer"                    "character"                    "character"
                     company             days_in_waiting_list                  customer_type                            adr
                   "character"                      "integer"                    "character"                      "numeric"
  required_car_parking_spaces        total_of_special_requests             reservation_status          reservation_status_date
                   "integer"                      "integer"                    "character"                    "character"
```

La función “column_info” está diseñada para proporcionar información sobre la estructura de un dataframe en R. Al llamar a esta función con un dataframe como argumento, imprimirá los nombres de las columnas seguidos de los tipos de datos de cada columna. Esto te permite obtener una rápida visión general de la composición del dataframe, incluyendo qué variables están presentes y qué tipo de datos contienen. Es una herramienta útil para comenzar a explorar y comprender tus datos en R.

- Función para mostrar los 10 primeros valores de cada columna:

```
inspect_unique_values <- function(hotels) {
  cat("Valores únicos en cada columna (primeros 10):\n")
  print(sapply(hotels, function(x) head(unique(as.vector(x)), 10)))
}
```

```
inspect_unique_values(hotels)
```

```
Valores únicos en cada columna (primeros 10):
$hotel
[1] "Resort Hotel" "City Hotel"

$is_canceled
[1] 0 1

$lead_time
 [1] 342 737   7  13  14   0   9  85  75  23

$arrival_date_year
[1] 2015 2016 2017

$arrival_date_month
 [1] "July"      "August"    "September" "October"   "November"  "December"  "January"   "February"  "March"     "April"

$arrival_date_week_number
 [1] 27 28 29 30 31 32 33 34 35 36

$arrival_date_day_of_month
 [1]  1  2  3  4  5  6  7  8  9 10

$stays_in_weekend_nights
 [1]  0  1  2  4  3  6 13  8  5  7

$stays_in_week_nights
 [1]  0  1  2  3  4  5 10 11  8  6

$adults
 [1]  2  1  3  4 40 26 50 27 55  0

$children
[1]  0  1  2 10  3 NA

$babies
[1]  0  1  2 10  9

$meal
[1] "BB"        "FB"        "HB"        "SC"        "Undefined"

$country
 [1] "PRT"  "GBR"  "USA"  "ESP"  "IRL"  "FRA"  "NULL" "ROU"  "NOR"  "OMN"

$market_segment
[1] "Direct"        "Corporate"     "Online TA"     "Offline TA/TO" "Complementary" "Groups"        "Undefined"     "Aviation"

$distribution_channel
[1] "Direct"    "Corporate" "TA/TO"     "Undefined" "GDS"

$is_repeated_guest
[1] 0 1

$previous_cancellations
 [1]  0  1  2  3 26 25 14  4 24 19

$previous_bookings_not_canceled
 [1] 0 1 2 3 4 5 6 7 8 9

$reserved_room_type
 [1] "C" "A" "D" "E" "G" "F" "H" "L" "P" "B"
```

```
 [1] "C" "A" "D" "E" "G" "F" "I" "B" "H" "P"

$booking_changes
 [1]  3  4  0  1  2  5 17  6  8  7

$deposit_type
[1] "No Deposit" "Refundable" "Non Refund"

$agent
 [1] "NULL" "304"  "240"  "303"  "15"   "241"  "8"    "250"  "115"  "5"

$company
 [1] "NULL" "110"  "113"  "270"  "178"  "240"  "154"  "144"  "307"  "268"

$days_in_waiting_list
 [1]   0  50  47  65 122  75 101 150 125  14

$customer_type
[1] "Transient"       "Contract"        "Transient-Party" "Group"

$adr
 [1]   0.0  75.0  98.0 107.0 103.0  82.0 105.5 123.0 145.0  97.0

$required_car_parking_spaces
[1] 0 1 2 8 3

$total_of_special_requests
[1] 0 1 3 2 4 5

$reservation_status
[1] "Check-Out" "Canceled"  "No-Show"

$reservation_status_date
 [1] "2015-07-01" "2015-07-02" "2015-07-03" "2015-05-06" "2015-04-22" "2015-06-23" "2015-07-05" "2015-07-06" "2015-07-07" "2015-07-08"
```

Primero, imprime un mensaje indicando que va a mostrar estos valores únicos. Luego, utiliza la función sapply() para aplicar una función anónima a cada columna del dataframe. Esta función anónima toma cada columna, encuentra los valores únicos en ella (unique(x)), selecciona los primeros 10 valores (head(unique(x), 10)), y luego los imprime.

- **<u>Procesar datos:</u>**

-Identificación de datos faltantes (NA)

```
extraer_nombres_y_codigos <- function(datos) {
  # Extrae los primeros 10 nombres de hoteles
 primeros_10_hoteles <- head(datos$hotel, 10)

  # Extrae los primeros 10 códigos de compañías
 primeros_10_companias <- head(datos$company, 10)

  # Crea un data frame con los resultados
 resultados <- data.frame(Hotel = primeros_10_hoteles, Compañía =
primeros_10_companias)

 return(resultados)
}

# Llama a la función con tu conjunto de datos 'hotels'
resultados <- extraer_nombres_y_codigos(hotels)

# Imprime los resultados
print("Nombres de los primeros 10 hoteles y códigos de las primeras 10 compañías:")
print(resultados)
```

```
[1] "Nombres de los primeros 10 hoteles y códigos de las primeras 10 compañías:"
> print(resultados)
          Hotel Compañía
1  Resort Hotel      NULL
2  Resort Hotel      NULL
3  Resort Hotel      NULL
4  Resort Hotel      NULL
5  Resort Hotel      NULL
6  Resort Hotel      NULL
7  Resort Hotel      NULL
8  Resort Hotel      NULL
9  Resort Hotel      NULL
10 Resort Hotel      NULL
```

La función extraer_nombres_y_codigos extrae los nombres de los primeros 10 hoteles y los códigos de las primeras 10 compañías de un conjunto de datos. Luego, crea un nuevo dataframe con estos resultados y lo devuelve. Finalmente, se llama a la función con un conjunto de datos llamado hotels y se imprimen los resultados.

-Explicación y aplicación de la técnica utilizada para eliminar o completar los datos faltantes

```
# Convierte los valores en la columna 'company' a enteros
```

```
hotels$company <- as.integer(hotels$company)
```

```
# Imprime la columna 'company' actualizada
```

```
print("Columna 'company' actualizada como enteros:")
```

```
print(hotels$company)
```

```
[1] "Columna 'company' actualizada como enteros:"
> print(hotels$company)
  [1]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 110  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
 [40]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
 [79]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[118]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[157]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[196]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 110 110 110  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[235]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[274]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 113  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[313]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[352]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 270  NA  NA
[391]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[430]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 178  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[469]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[508]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 240  NA  NA  NA
[547]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[586]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 154  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[625]  NA  NA  NA 178  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[664]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[703]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[742]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[781]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 270  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[820]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[859]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[898]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA 144
[937]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
[976]  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
# Reemplaza los valores nulos en la columna 'company' con números aleatorios
entre 50 y 300
```

hotels$company[is.na(hotels$company)] <- sample(50:300, sum(is.na(hotels$company)), replace = TRUE)

```
# Imprime la columna 'company' actualizada
```

print("Columna 'company' actualizada con valores aleatorios:")

print(hotels$company)

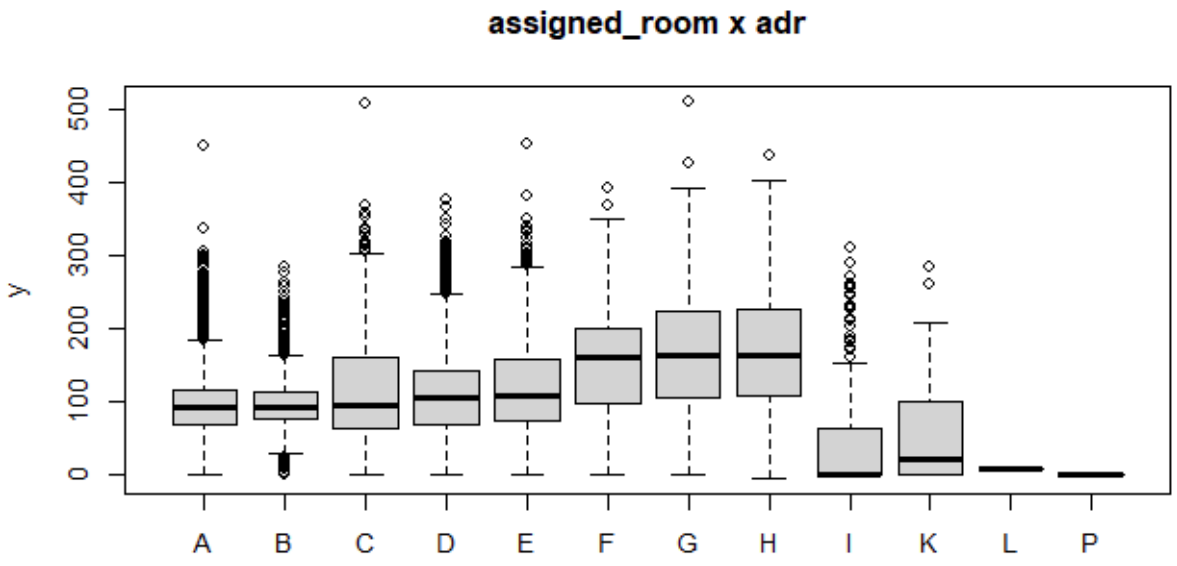
```
[1] "Columna 'company' actualizada con valores aleatorios:"
> print(hotels$company)
   [1] 234 217 140 269 266 295 146 197 111 151 190 132 211 213 116  83 297  85 110 205 160 106 155 199 123 296 109 146 117 221  58 140 275 108 109 228 241  73 115
  [40] 112 123  62 216 104 299 292 133  97 141  77 251  55 115 182 255 184 148 116  60 271 168 104 196 252 125 148 279 244 181 221 194 217 167 115 259 242 295  63
  [79]  76 285 144 291 276 255 272 229 205 156 187 100  98 130 281 246 216 199  63 204  65  93 219  88 122 182  85 139 212 242 128 186  94 102 273 264 178 134 183
 [118] 140 198 149 118 109  64 241 226 176  70  84 111  78 269  76 273 185 105 133 298 141 144 113 272  99 240 233  54 102  68 110 165 273  50 141 244 234 269 166
 [157] 282 107 107 195  75 221 236 222 242  72  89 108 281  96 144  60 173 151 193 263 253  81 195 164 288  97 269 281 143 186 281 130 223  65 190 170 181 201  51
 [196] 198 287 163  75  90  59  53 199 197  69 245 159 110 280 196 147  93 166  63 100 159 105 135 110 110 110  85 200 144  52 264 137 241 134 230 251  79 254 293
 [235] 260 155 109 264 225 157 269 254 141 196 192  67 279  73  50 146 206  76 158 189 165 251 115 233 246 254 126 221 107 292  55 249 188 206 154 216 203  62 299
 [274]  65  77 219 134 212 152 116  89 161 132 197 274 227 176 202 210 260 106  52 275 127 179 113  92 166 139 191 258  52 141  68 291 283 219 112  61  97  87 112
 [313] 225  88 270 236 152  87 108 216 242 280  83  87 207 256 116 166  81 224 259 101  61 165  53  73 170 156 219  68  77 245 142 229 124 195  67  69 227 245  56
 [352]  74 258 282 227  62 270 233 278 286  84 228  58  63  51  88 289 180 129  60 252 220 109 175 253 147 114  73  85 165 285 140 284 233 118 109 268 270 122 118
 [391] 197 245  92 222 202 193  71 133  60 125  90 162 117 166 167 269 159 250  77 123  54  55 163 231 141 174  97 118 203 243 179 293 287  50 136 130 292 105 208
 [430] 300  50 246 284 163  52 223  80 150  93 237  54 118 264 159 200 224 126 151  71 292 178  53 198 178 127 267 194 188 291 211  90 113 156 232 270 284 294 295
 [469] 222 143  56 107 100 203 274 247  93 182 102 195 174 268 292 127 190 299 144 261 149 173 229 149 277 124 236 208 299  66 255 183 173  82  89 221  67  70 125
 [508] 220 147 110 155 203 133  52 209 253 201 248 196  93 208 102  94 152 209 280 255  53 217 169 196 255 187 150 164 289 137 298  78  94 245 240 124 228 295 214
 [547] 137 300 141 233 228 172 209  95  86 246 260 207 268  94 135 193 261  96 244  98 194 242 150 161 136 288 254  66 154  54 111 202 203 149 122 196  61 192 258
 [586]  88  52 111  82 249 251 177 160  81 245 106 223 244 174 212 154 220 118  91 263 193 160 206 170 205 107 135 207 128 278 234 190 201 104 237  84  91 145  69
 [625] 149  88 261 190 178  85 154 207 238 172 217 164 119  82 276 127 269 141 233 149 271  99 108 156 282  73 187 197 137  92 175 192 167 125 159 105  57  88  60
 [664] 204 164 131 237 235 213  75  69 136 104 145 171 112 299  65 211  76 146 191 140 190  53  53 268 226 110 277 145 153  73 114 181 101 162 123 235 108 292 177
 [703]  78 159 238 298 294  64 157 155 102  92 146 105 133  74 295 185 103 134  97 256 289 264 174 186 244 248 258 192 194 291 253 239  58 102 149 224 111 250 199
 [742]  82 218 182 241 211  87 165 286 152 102 165  66 123 138  72 106  75 199 211 235 178 139 121 244 215  62 140 235  53 116  65 171 127 108  91 108 221  52  72
 [781] 148 184 113 128  91  80 223 100  64 150 289 141 105 242  64 179 270 193  83 262 202 297 272 159 128  73 136 121 199 113 115 161  84 291 207  80 209 163 162
 [820] 233 194 172  68 228  73 157 244 192 206 296  84 171 229  98  68 234 133 292 223 210  80 188 213 177  51 153 145 105 280 153 159 250 169 239 298 227 229 181
 [859] 188 128 140  60 214 217  63  57 130  99 231  63  76 199 123  99 218 285 284  99 222 149 166 163 232  76 122 246 207 203 236 141 275 129 148  58 127 295 271
 [898] 150 271  58 108 242  58 135 114  91 277 127 141 125  57 181 236 143 205 111  60 252  97 221 111 226 126 163 154 126 107 260 268 231  55 190 145 107 284 144
 [937] 214 101 108  52  64  67 300 204 134 219 116  66  87 280  51 197 152 116 274 203 171 188 224 206 256 127 131 272 165 291  88 255 217  72 144  70  95 238 240
 [976] 214 203 136 131 114  80 256 134  76 132 225 241 118 227  51  72 200 278 111 114 135 173 242  99  54
```

En primer lugar, asumimos que ya has cargado un conjunto de datos llamado ‘hotels’ en un marco de datos. La columna ‘company’ contiene códigos de compañías asociadas a las reservas de hotel. Para reemplazar los valores nulos (NA) en esta columna, generamos números aleatorios entre 50 y 300 utilizando la función sample(50:300, sum(is.na(hotels$company)), replace = TRUE). Luego, asignamos estos valores aleatorios a los lugares donde encontramos valores nulos en la columna ‘company’. Finalmente, convertimos todos los valores en la columna ‘company’ a enteros utilizando as.integer(hotels$company). El resultado final es una columna ‘company’ actualizada con valores aleatorios y enteros.

- Explicación y aplicación de la(s) técnica(s) utilizada(s) para transformar los datos atípicos.

```
# La columna 'assigned_room_type' contiene los tipos de habitación asignados
(por ejemplo, 'A', 'B', 'C', etc.).

# La columna 'adr' representa el precio promedio diario.
```

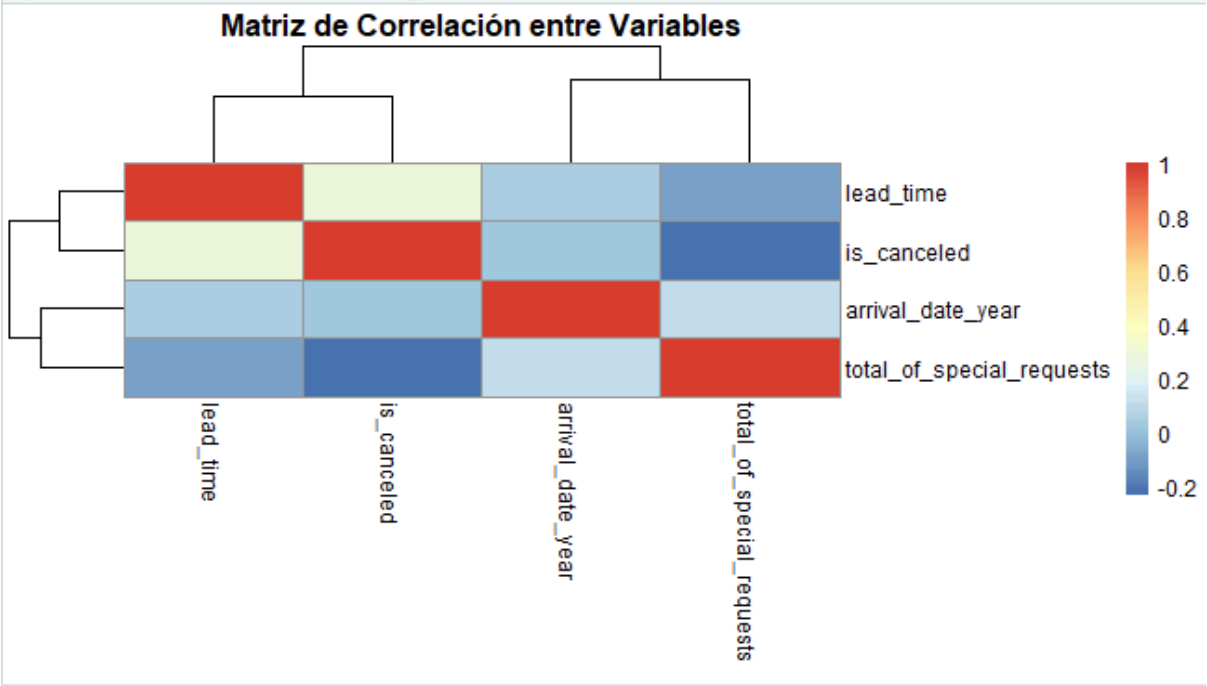
```
# Paso 1: Conversión de la columna 'assigned_room_type' a factor

hotels$assigned_room_type <- as.factor(hotels$assigned_room_type)

# Paso 2: Creación de un gráfico de dispersión original

plot(hotels$assigned_room_type, hotels$adr)

# Este gráfico muestra la relación entre los tipos de habitación asignados y
los precios promedio diarios.

# Paso 3: Filtrado de datos para valores de 'adr' menores a 5000

HB2 <- hotels %>% filter(hotels$adr < 5000)

# Paso 4: Creación de un segundo gráfico de dispersión con datos filtrados

plot(HB2$assigned_room_type, HB2$adr, main = "assigned_room x adr")

# Este gráfico muestra la relación entre los tipos de habitación asignados y
los precios promedio diarios,

# pero solo para aquellos registros con precios menores a 5000.
```



En el código proporcionado, se trabaja con un conjunto de datos llamado ‘hoteles’. Primero, se convierte la columna ‘assigned_room_type’ en un factor para representar los diferentes tipos de habitaciones asignadas a los huéspedes en un hotel. Luego, se crea un gráfico de dispersión para visualizar la relación entre estos tipos de habitaciones y el precio promedio diario (adr). Posteriormente, se filtran los datos para incluir sólo aquellos registros con precios de habitación menores a 5000. Finalmente, se genera un segundo gráfico de dispersión utilizando los datos filtrados, enfocándose en los precios más bajos. El objetivo es explorar cómo los diferentes tipos de habitaciones afectan los precios promedio en el contexto hotelero.

- **Visualizar datos:**

```
# Selecciona las variables relevantes (puedes agregar más según tus
necesidades)

selected_vars <- hotels[, c('lead_time', 'is_canceled', 'arrival_date_year',
'total_of_special_requests')]

# Calcula la matriz de correlación

cor_matrix <- cor(selected_vars)

# Instala y carga el paquete 'pheatmap' si aún no lo has hecho

# install.packages('pheatmap')

library(pheatmap)

# Crea un mapa de calor para visualizar las correlaciones

pheatmap(cor_matrix, scale = 'none', main = "Matriz de Correlación entre Variables")
```



1. Correlaciones entre variables:
    - lead_time (Número de días entre la reserva y la llegada) y is_canceled (Si la reserva fue cancelada):
        - Existe una correlación positiva entre lead_time y la probabilidad de cancelación (is_canceled). Esto sugiere que cuanto mayor es el tiempo de anticipación para la reserva, es más probable que la reserva se cancele.
        - total_of_special_requests (Número de solicitudes especiales realizadas por el huésped) y is_canceled:
        - Existe una correlación negativa entre el número de solicitudes especiales y la probabilidad de cancelación. Esto significa que los huéspedes que hacen más solicitudes especiales tienen menos probabilidades de cancelar su reserva.
    - arrival_date_year (Año de llegada) y lead_time:
        - No hay una correlación clara entre el año de llegada y el tiempo de anticipación para la reserva. Esto sugiere que el año no afecta significativamente la planificación anticipada de las reservas.
2. Inferencias:

- Los huéspedes que reservan con mucha antelación (mayor lead_time) pueden ser más propensos a cancelar, posiblemente debido a cambios en sus planes o circunstancias.
- Las solicitudes especiales parecen influir en la decisión de cancelación. Los huéspedes que hacen más solicitudes especiales pueden estar más comprometidos con su reserva y menos propensos a cancelar.
- El año de llegada no parece ser un factor importante en la planificación anticipada de las reservas.

3. Recomendaciones:
    - Para reducir las cancelaciones, el hotel podría considerar:
        - Ofrecer incentivos a los huéspedes que reservan con mucha antelación.
        - Personalizar las ofertas para aquellos que hacen solicitudes especiales.
        - Monitorear las tendencias de cancelación a lo largo del tiempo y ajustar las estrategias en consecuencia

V. Conclusiones

- La limpieza de dataset es un paso importante para poder recuperar valores que están encubiertos por un “NA”, también se puede usar el método de eliminación de columnas ya que no son datos relevantes para el análisis del dataset.

- Se observa que la gente prefiere el tipo "City Hotel" en comparación con "Resort Hotel". Esto se evidencia por el mayor número de reservas realizadas en City Hotel en comparación con Resort Hotel.

- Después de un aumento en la demanda hasta 2016, se observa una tendencia a la baja en las estancias en fines de semana a partir de entonces. Esto puede indicar un cambio en las preferencias de los clientes o cambios en el mercado que podrían afectar la demanda.

- Se identifican tres temporadas: baja, media y alta. Los meses de temporada baja son enero, noviembre y diciembre. La temporada alta abarca julio y agosto, mientras que el resto de los meses se consideran de temporada media.

- Un total de 9336 reservas incluyen niños o bebés. Esto sugiere que hay una demanda considerable de alojamientos que puedan acomodar a familias.

- Se observa que las cancelaciones de reservas son más frecuentes cuando no hay espacios de estacionamiento disponibles. Esto indica que contar con espacios de estacionamiento puede influir significativamente en la decisión de reserva de los clientes.

- Para futuros análisis, una recomendación precisa es usar la función pasar variables a factor para se puedan hacer reportes o resúmenes de las tablas del dataset, por el contrario lo leería como un “character” y por ende no te daría los datos que buscas de forma ordenada.

- Como lección nos queda que todo reporte de datos, nos sirve poder gestionar de una manera correcta y precisa el análisis de los futuros datasets a utilizar. Además la creación de una matriz de correlación de variables, es de vital importancia ya que sirve para entender la naturaleza y la fuerza de las relaciones lineales entre variables en un conjunto de datos.

VI. Bibliografía

- Capozzi, L. C., Barresi, A. A., & Pisano, R. (2019). Supporting data and methods for the multi-scale modelling of freeze-drying of microparticles in packed-beds. *Data in Brief*, *22*, 722–755. https://doi.org/10.1016/j.dib.2018.12.061